Towards AER VITE: building spike gate signal

Fernando Perez-Peña, Arturo Morgado-Estevez and Carlos Rioja-Del-Rio
Applied Robotics Lab, High School of Engineering
University of Cadiz, Spain
fernandoperez.pena@uca.es

Alejandro Linares-Barranco, Angel Jimenez-Fernandez
Department of Computer Architecture and Technology
ETSI Computer Science
University of Seville, Spain

Juan Lopez-Coronado, Jose Luis Muñoz-Lozano
Automatic and System Engineering Department
University Polytechnics of Cartagena, Spain

***Abstract*—Neuromorphic engineers aim to mimic the precise and efficient mechanisms of the nervous system to process information using spikes from sensors to actuators. There are many available works that sense and process information in a spike-based way. But there are still several gaps in the actuation and motor control field in a spike-based way. Spike-based Proportional-Integrative-Derivative controllers (PID) are present in the literature. On the other hand, neuro-inspired control models as VITE (Vector Integration To End point) and FLETE (Factorization of muscle Length and muscle Tension) are also present in the literature. This paper presents another step toward the spike implementation of those neuro-inspired models. We present a spike-based ramp multiplier. VITE algorithm generates the way to achieve a final position targeted by a mobile robotic arm. The block presented is used as a gate for the way involved and it also puts the incoming movement on speed with a variable slope profile. Only spikes for information representation were used and the process is in real time. The software simulation based on Simulink and Xilinx System Generator shows the accurate adjust to the traditional processing for short time periods and the hardware tests confirm and extend the previous simulated results for any time. We have implemented the spikes generator, the ramp multiplier and the low pass filter into the Virtex-5 FPGA and connected this with an USB-AER (Address Event Representation) board to monitor the spikes.**

I. INTRODUCTION

Since the very beginning, it is known that nervous systems use spikes or graduated potentials to send information across the body. The excellent behavior of these systems leads us to mimic them into electronic devices based on interconnected neuron systems. Nevertheless, since neurons communicate in a point-to-point manner and it is possible to integrate several thousands of artificial neurons into the same electronic device (VLSI chip or FPGA), new communication strategies were taken, like the Address-Event-Representation (AER) protocol [1]. AER maps each neuron with a fixed address which is transmitted through the interconnected neuron system using a simple and fast handshake protocol (with parallel buses) or flow control protocol (in high-speed serial buses) solving the communication problem between neuromorphic chips.

By using AER protocol, all neurons are continuously sending information about their excitation level to the central system and thus it could be processed in real time by a higher layer. AER is based on the concept of mimicking the structure and information coding of the brain. Thus AER let us process the information in real time. This is one of its interesting features: the speed provided. Another relevant characteristic is the scalability that allows this by parallel connections.

Currently, there is a small but growing number of research groups worldwide working on the AER technology premises, and so many works could be listed. On the one hand, VLSI chips for sensors [2-4]; on the other hand, extended systems are spike-based PID motor controller [5], neuro-inspired robotics [6] and bio-inspired systems for processing, filtering or learning [7-10].

Taking the advance of the previous works [11] our motivation is to integrate the visual information from an AER retina with the spikes control into a bio-inspired robotic arm by using solely spikes for the whole process. Current research is focused on two layers, generating the trajectory and applying it to the mimic muscles by motors. The second layer would be the next step when the previous one finished. There are several algorithms for both tasks. Nevertheless, we have chosen a pair of algorithms developed by Daniel Bullock and Stephen Grossberg [12-13]. They are based on classic neural networks and they fit with our target. References [14] and [15] use the VITE algorithm, although at its original form including the complex processing we are going to try to avoid.

In this work we present a neuro-inspired block that solves a spike-based ramp operation needed for neuro-inspired motor control algorithm. This block will be excited with streams of pulses (spikes) at the output and so only spikes should be

This work was supported by the Spanish grant VULCANO (TEC2009-10639-C04-02).

found along the system. There are several spike-based information codifications and the one we use is the rate coded one [16], when excitation is low, spike rate is low and thus the time between spikes is high; however, when signal excitation increases, the inter-spikes interval time (ISI) decreases, while spike rate increases.

II. THE FIRST LAYER: VITE ALGORITHM

The VITE (Vector Integration To End point) algorithm, assigned for the first layer, is shown in Fig.1. It calculates a non-planned trajectory by computing the difference between the target and the present position. It also introduced the problem to deal with different reference systems: one is for the visual sensor, another one for the central processing (typically the head) and the last one for the actuator. The way it solves this problem is using the motor references for all the system.

This algorithm introduced a non-specific control signal called 'GO'. This signal lets us separate the spatial pattern characteristics, such as distance and direction, and the energy of the movement. Thus it allows controlling the movement rate and gate.

References [17] and [18] justify this algorithm. They show by means of electromyogram how the activity is present at any area of the motor cortex before the muscles initiate the movement. Also the asymmetric speed profile obtained with movements is mentioned, as it seems to happen with the VITE algorithm. This concept highlights the importance of the GO signal and the results presented in this paper.

Approaching the block presented, first have a look into the VITE in Fig. 1. Afterwards, the details of the block will be described. The algorithm generates a non-planned trajectory. The GO signal is the gate for the movement. Until it is nonzero, the motor priming occurs by computing the difference vector. On its own, it has no effect. It needs the second layer to apply forces to the muscles.

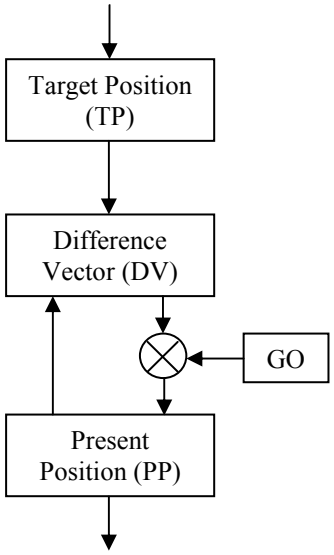

Figure 1. Block diagram of the VITE algorithm. The target position will be supplied by an AER vision sensor and the present position calculated at each time will be supplied by the second layer.

In its simplest form, the VITE circuit obeys the equations:

$$\frac{d}{dt}\,DV = \propto \times (TP - PP - DV), \quad (1)$$

$$D/dt\ PP = GO \times DV. \quad (2)$$

According to (1) and (2), and regardless of the GO signal to apply the classic control theory, the result is shown in Fig. 2 above. DV is calculated by a low pass filter to avoid the abrupt movements. All the blocks gotten are solved at [19] and their translation is automatic as it is shown in Fig. 2 below.

III. FOCUSING ON THE BLOCK: RAMP MULTIPLIER

Now we must deal with the GO signal. Remember it should multiply the DV signal to put it on speed and be the gate of the movement. There are several types of signals they could carry with this task but we chose the ramp signal. It is the easiest way to perform the desired behavior at the spikes paradigm.

If we think according to the spike-based information codified, the multiplier block will need to inject spikes in accordance with the slope value of the ramp every time a spike is fired by the previous block.

To implement it, a *slope counter* is designed. At its fixed value, the relative slope increases in one unit until the saturation value is reached. The general slope is calculated as (3). The number of bits used to synthesize the *slope counter* limits the global value of the slope. We need a 24-bit in order to achieve a value of 16,777,215 clock cycles resulting a 0.3355 seconds per slope unit. Thus, a slope of approximately 3 percent will be the minimum for the total of bits.

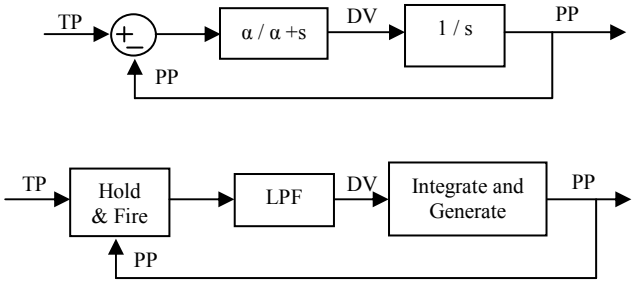

Figure 2. Above: Block diagram resulted from the conversion of the VITE algorithm to Laplace domain. Below: Block diagram generated from existing spikes processing blocks. The Spike Hold & Fire block performed the subtraction between the present position and the target position; both signals are spike streams. The Spikes Integrate & Generate block allows us to integrate the DV signal (again a spike stream). This block is composed by a spike counter and a spike generator. The latter uses a parameter called IG_FD (Integrate&Generate_FrequencyDivider) to divide the clock signal and generate the output stream according to this division. The GO signal does not appear because it has not been considered to be able to apply the classic control theory but it will be present at the input of the integrate and generate block to put on speed the DV signal.

IV. SCENARIO SIMULATION

It consists of a spike generator, the ramp multiplier and a low pass filter to redistribute the spikes. The spike frequency of the generator and the multiplier can be calculated as expressed in (4) and (5). Each input is for each block. The cut-off frequency is calculated as (6) and it has unitary gain like it is expressed in Fig. 2 above. Notice that NBITS is a constant structure which value is 16 and the clock frequency is 50 MHz for all the tests performed. Also, a value of 242.85 Hz is calculated with IG_FD zero considered. Furthermore, the multiplier spike frequency varies with time so it should be multiplied by time to calculate the instantaneous frequency. The achievable slope compared to bits used is shown in Table 1.

$$\text{Slope} = {}^{f_{clk}}/_{\text{slope}_{counter}}, \tag{3}$$

$$f_{generator} = \left({}^{f_{clk}}/_{2^{NBITS-1}}\right) \times \text{input}, \tag{4}$$

$$f_{multiplier} = (f_{clk}/\text{slope}_{counter}) \times \text{input}, \tag{5}$$

$$\omega_{cut-off} = {}^{f_{clk}}\Big/_{(2^{NBITS-1}) \times (IG_{FD}+1)}. \tag{6}$$

TABLE I. MINIMUM VALUES FOR THE SLOPE VS. NUMBER OF BITS USED FOR THE SLOPE COUNTER.

Bits Vs. Min. Slope	
Bits used	***Min. Slope***
8	196078.43
16	762.95
24	2.98
32	0.0116

V. SOFTWARE SIMULATION RESULTS

With simulation software Simulink by Matlab and Xilinx System Generator, a suitable environment was developed to perform the tests.

The effect of the filter can be observed in Fig. 3 and Fig. 4. Also in Fig. 3, the latency at the beginning is shown. It is due to the clock slope counter which increases the slope. For this graph, the clock is 50 MHz and the increasing slope counter is 8192 clock cycles (0.1638 ms per each increment at the slope, which means a slope of approximately 6103). Fig. 4 shows the effect of increasing IG_FD with the consequent decrease of cut-off frequency.

All the tests performed with the software platform were time limited due to the huge data processing carried out by the computer. Also, this leads the slope to only take a higher value that allows the slope counter raising its value an increase of almost one unit to check the behavior. This is the reason why we moved to the hardware scenario.

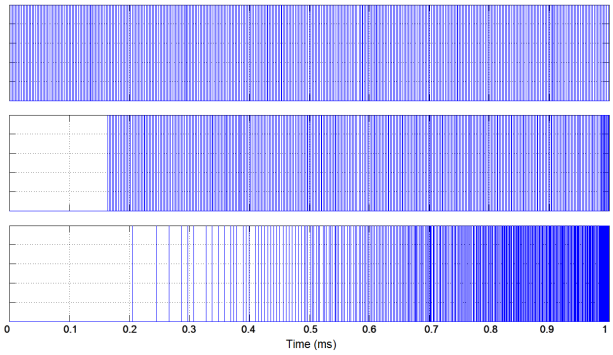

Figure 3. Spike details from the software scope provided by Simulink. The first row shows the spike input to the system, the second shows the output from the multiplier block and the third one shows how the low pass filter redistributes the equidistant spikes from the output of the previous block.

VI. HARDWARE TESTS RESULTS

The scenario is composed by a Virtex-5 FPGA and a USB-AER board [20] to monitor the spikes. The results obtained with hardware are shown in Fig. 5 and Table 2 shows the data used for each test. The slope for each test should be calculated with (3).

With the previous simulation shown, it was not necessary to saturate the maximum of the ramp because there were no software limitations. That is the opposite for hardware tests. These tests would be limited by the monitor board and its maximum spike rate, fixed at 5 Mevps (Mega events per second). The saturation level for the ramp was fixed at 20. It means any input will be multiplied as maximum by twenty.

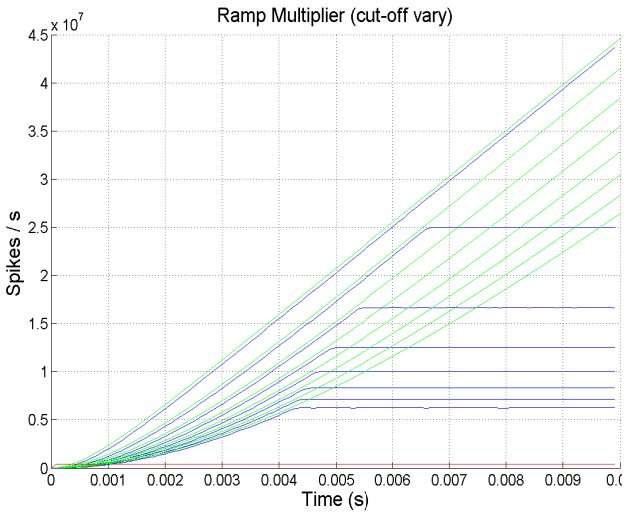

Figure 4. Software simulation results for the ramp multiplier block when the IG_FD parameter varies between [0-7]. The red line is the spike frequency from the input to the system. The green lines are the theorical behavior and the blue lines are the simulation results. The input for the generator was 255. This means an output frequency of 390 Kevents / s. The accurate adjustment to the theorical behavior is shown.

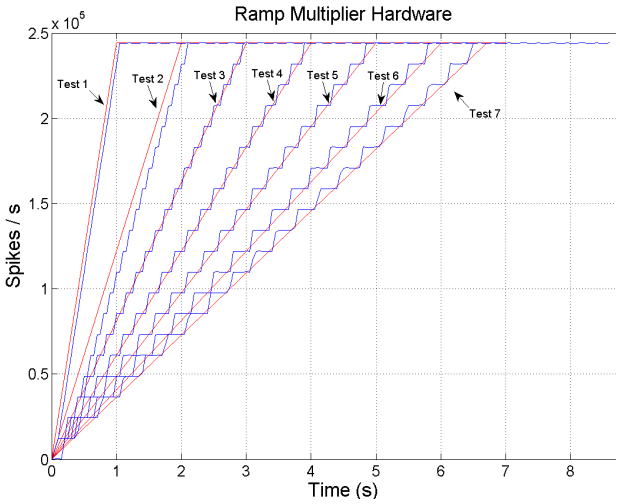

Figure 5. Results for hardware tests are shown. The red line represents the theorical behavior and the blue line represents the hardware results. The input for the generator was 8. It means an output frequency of 12,207 Kevents / s and multiplied by 20 in saturation it yields a total of 244,14 Kevents / s reached at different times at each test. The table below shows the parameters for the test. They are matched by the test numbers.

TABLE II. DATA USED FOR HARDWARE TESTS

Test No.	**Tests performed data**			
	Slope counter	***Slope***	***Second per slope unit***	***Time to saturation (seconds)***
1	2,5 M	20	0.05	1
2	5 M	10	0.1	2
3	7,5 M	6.66	0.15	0.33
4	10 M	5	0.2	4
5	12,5 M	4	0.25	5
6	15 M	3.33	0.3	6
7	2^{24} -1 M	2.98	0.33	6.71

VII. CONCLUSIONS

In this work we have presented a spike-implemented GO signal in order to use it within the VITE algorithm. It was tested within a simulation scenario and synthesized into a digital device. Both tests have demonstrated the fine adjustment to the theoretical behavior. A 24-bit counter was designed to perform a three percent slope as minimum. If a unitary slope on a ramp is needed, then a 26-bit counter should be implemented. At the simulation tests there was no limit at the spike rate compared to the real device, where we were limited by the spikes monitor board.

REFERENCES

[1] Sivilotti, M. Wiring Considerations in Analog VLSI Systems with Application to Field-Programmable Networks. Ph.D. Thesis, California Institute of Technology, Pasadena, CA, USA, 1991.

[2] Lichtsteiner, P., Posch, C., Delbruck, T., "A 128 × 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," IEEE J. Solid-State Circuits 2008, vol. 43, pp. 566–576.

[3] Chan, V., Liu, S.C., van Schaik, A., "AER EAR: A matched silicon cochlea pair with address event representation interface," IEEE Trans. Circuits Syst. I: Regul. Pap. 2007, vol. 54, pp. 48–59.

[4] Liu, S.C., van Schaik, A., Minch, B.A., Delbruck, T, "Event-based 64-channel binaural siliconcochlea with Q enhancement mechanisms," in Proceedings of the 2010 IEEE InternationalSymposium on Circuits and Systems (ISCAS), Paris, France, June 2010, pp. 2027–2030.

[5] Jimenez-Fernandez, Angel; Jimenez-Moreno, Gabriel; Linares-Barranco, Alejandro; Dominguez-Morales, Manuel J.; Paz-Vicente, Rafael; Civit-Balcells, Anton. 2012. "A Neuro-Inspired Spike-Based PID Motor Controller for Multi-Motor Robots with Low Cost FPGAs.," Sensors 12, no. 4, pp. 3831-3856.

[6] Linares-Barranco, A., Gomez-Rodriguez, F., Jimenez-Fernandez, A., Delbruck, T., Lichtensteiner, P., "Using FPGA for visuo-motor control with a silicon retina and a humanoid robot," in Proceedings of ISCAS 2007IEEE International Symposium on Circuits and Systems, New Orleans, LA, USA, 27–30 May 2007, pp. 1192–1195.

[7] Barranco, F., Diaz, J., Ros, E., del Pino, B., "Visual system based on artificial retina for motion detection," IEEE Trans. Syst. Man Cybern. Part B: Cybern. 2009, vol. 39, pp. 752–762.

[8] Hafliger, P., "Adaptive WTA with an analog VLSI neuromorphic learning chip," IEEE Trans. Neural Netw. 2007, vol. 18, pp. 551–572.

[9] Indiveri, G., Chicca, E., Douglas, R., "A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," IEEE Trans. Neural Netw. 2006, vol. 17, pp. 211–221.

[10] Chicca, E., Indiveri, G., Douglas, R.J., "An event-based VLSI network of integrate-and-fire neurons," in Proceedings of 2004 International Symposium on Circuits and Systems (ISCAS '04), Vancouver, Canada, 23–26 May 2004, vol. 5, pp. 357–560.

[11] Linares-Barranco, A., Paz-Vicente, R., Jimenez, G., Pedreno-Molina, J.L., Molina-Vilaplana, J., Lopez-Coronado, J., "AER neuro-inspired interface to anthropomorphic robotic hand," in Proceedings of International Joint Conference on Neural Networks, Vancouver, BC, Canada, 16–21 July 2006, pp. 1497–1504.

[12] Bullock, D. and Grossberg, S., "Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation," Psychological Review, vol. 95, pp. 49–90, 1988.

[13] Bullock, D. and Grossberg, S., "The VITE model: A neural command circuit for generating arm and articulator trajectories," in J.A.S. Kelso, A.J. Mandell, and M.F. Shlesinger(Eds.), Dynamic Patterns in Complex Systems. Singapore: World Scientific Publishers, pp. 305–326, 1988.

[14] Vilaplana J.M., Coronado J.L., "A neural network model for coordination of hand gesture during reach to grasp,". Neural Networks, IEEE Transactions on, vol.19, no.1, pp 12-30, May 2006.

[15] Pattacini U., Nori F., Natale L., Metta G., and Sandini G., "An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots," in The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010.

[16] Linares-Barranco, A., Jimenez-Moreno, G., Linares-Barranco, B., Civit-Balcells, A., "On algorithmic rate-coded AER generation," Neural Networks, IEEE Transactions on , vol.17, no.3, pp.771-788, May 2006.

[17] Georgopoulos, A.P., "Neural integration of movement: role of motor cortex in reaching," The FASEB Journal, vol 2, no 13, pp. 2849-2857, October 1988.

[18] Nagasaki, H., "Asymmetric velocity and acceleration profiles of human arm movements," Experimental Brain Research, vol 74, no 2, pp 319-326, 1989.

[19] Jiménez-Fernández, A., "Diseño y evaluación de sistemas control y procesamiento de señales basadas en modelos neuronales pulsantes," Ph.D. Thesis, Universidad de Sevilla, Sevilla, Spain, 2010.

[20] Berner, R., Delbruck, T., Civit-Balcells, A., Linares-Barranco, A., "A 5 Meps $100 USB2.0 address-event monitor-sequencer interface," in Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS , New Orleans, LA, USA, 27–30 May 2007, pp. 2451–2454.